

# Model Predictive HVAC Control with Online Occupancy Model

Justin R. Dobbs\*, Brandon M. Hency

*Department of Mechanical and Aerospace Engineering, Upson Hall, Cornell University, Ithaca, NY 14853, USA*

---

## Abstract

This paper presents an occupancy-predictive control method for heating, ventilation, and air conditioning (HVAC) systems in buildings that incorporates the building's thermal properties, local weather predictions, and a self-tuning stochastic occupancy model to reduce energy consumption while maintaining occupant comfort. Contrasting with existing approaches, its occupancy model requires no manual tuning and adapts to changes in occupancy patterns during operation. A prediction-weighted cost function provides pre-conditioning of thermal zones before occupancy begins and reduces system output before occupancy ends. Simulation results with conference room occupancy data demonstrate its effectiveness.

*Keywords:* model predictive control, MPC, occupancy prediction, on-line training, Markov chains, HVAC

---

## 1. Introduction

The objective of HVAC control systems is to maintain comfortable conditions for building occupants. Increasing energy prices have driven greater interest in demand-based control using schedules or occupancy triggering, but there are two problems with these approaches. First, fixed schedules become outdated; when occupancy patterns change, early or late occupants are left uncomfortable, or the space is conditioned prematurely or for too long. Second, thermal lag limits response speed and thus prevents aggressive temperature set-back in reactive systems. Addressing both schedule inaccuracy and thermal lag requires a stochastic occupancy model and a control scheme that can use its predictions effectively.

Considerable research effort has been directed toward occupancy detection and modeling. Detection research has largely focused on increasing accuracy through sensor fusion using probabilistic, neural, or sensor-utility networks [18, 4, 21, 16]. Agent-based models have been used to predict movement within buildings [19, 10], as have Markov chains [9, 23, 6]. Erickson and Dong, for example, modeled collections

---

\*Corresponding author.

*Email addresses:* jrd288@cornell.edu (Justin R. Dobbs), bmh78@cornell.edu (Brandon M. Hency)

of rooms as Markov states and movements among those rooms as transitions in order to predict the behaviors of individual occupants, while Dong and Lam [5] used a semi-Markov model to merge multiple sensor streams into an estimated occupant count. The simpler Page model considered the occupancy of a single zone with two states (occupied and vacant) under a time-heterogeneous Markov chain in order to create realistic occupancy patterns for building energy simulation, rather than for on-line prediction [23].

With the exception of the Page model, the above efforts have found use in heuristic [8, 11, 12] or model predictive control (MPC) schemes [5, 22, 12] but are bound by high implementation cost. For example, where authors have used MPC, they have used manually-generated thermal models [5, 22, 12] even though model creation is tedious and time-consuming. We instead employ an automated BIM translator outlined in a previous paper [13]. Furthermore, research has trended toward more complexity and adjustments—system topologies, control gains, and detection thresholds—resulting in “one-off” engineering efforts with favorable performance but without a clear path to widespread use. The system outlined in [5], for instance, uses CO<sub>2</sub>, sound, and light sensors that require carefully set detection thresholds for each room, plus an internal weather forecasting algorithm in lieu of commonly available on-line forecasts. We have removed unnecessary implementation decisions in favor of a simple occupancy prediction algorithm feeding predictions to an adaptation of standard MPC. The adjustments are few, and each serves a clearly-defined purpose. We have outlined every component’s operation with the practitioner in mind.

Second, recent research has paid little attention to on-line training. Although most training algorithms could be extended to work on-line, most studies have used off-line batch training without addressing ongoing maintenance [23, 5, 8]. A system which cannot train itself during operation will become out-of-date unless it is periodically retrained or can incrementally train itself with new observations. Our occupancy model is based on Page, et al [23] but adds on-line Bayesian inference to refine the occupancy model as new observations are made; the resulting system requires no batch training and maintains its performance without ongoing maintenance cost.

The paper progresses as follows. First, we outline the problem formulation and system architecture. Second, we describe the probabilistic occupancy model and its on-line training algorithm. Third, we discuss its integration with model predictive control. Finally, we present simulation results using actual occupancy data and compare our method’s performance to conservative scheduled and occupancy-reactive controllers. Throughout the discussion, the control scenario (heat only, small building, etc.) is kept deliberately simple

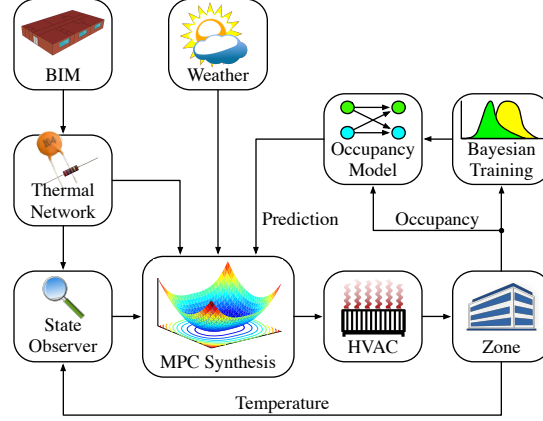


Figure 1: Proposed system architecture. For this study, the building model has been translated automatically from a CAD model into a linear, time-invariant network that encompasses the dominant thermal processes. (Model translation may also be performed manually.)

to emphasize the contribution of occupancy learning and prediction and its use with MPC.<sup>1</sup>

## 2. Problem Statement

We wish to minimize total energy usage of a building heating (or cooling) system while maintaining occupant comfort. Over conventional occupancy-reactive control, we aim to

- boost comfort at the beginning of occupancy by ensuring the space is conditioned when occupants arrive;
- reduce energy consumption before the beginning of occupancy by not conditioning the space prematurely; and
- exploit stored thermal energy toward the end of occupancy by switching off the system just before occupants leave.

Our approach, depicted in Figure 1, is based on MPC but uses a cost function weighted by occupancy predicted by a self-training stochastic model. At each step, the system measures the degree to which the space was occupied over the previous hour and uses the Markov chain to project future occupancy. The observation is then used to tune the Markov chain, which uses states  $I_{0...23}$  and  $O_{0...23}$  to denote occupancy

<sup>1</sup>See [12] for a comparison of MPC and heuristic control for a more complex HVAC system.

or vacancy for each hour of the day. The occupancy state, observed probabilistically with distribution  $\pi_k \in \mathbb{R}^{1 \times 48}$ , evolves according to

$$\pi_{k+1} = \pi_k P \quad (1)$$

where  $P \in \mathbb{R}^{48 \times 48}$  is an unknown matrix that we wish to estimate. The predicted occupancy and building temperature state are then used for MPC in which the controller finds a sequence of  $N$  future actions (heat inputs to the thermal zone) that minimize a total expected cost; this can be formulated as the convex optimization

$$\begin{aligned} & \text{minimize} && \sum_{j=0}^{N-1} \mathbb{E} [g(\tilde{x}_{j+k}, u_{j+k}, \Gamma_{j+k})] \\ & \text{subject to} && \tilde{x}_{i+1} = \tilde{A}\tilde{x}_i + B_u u_i + B_w \mathbb{E}[w_i] \quad \forall i \in \mathbb{Z}^+ \\ & && 0 \leq u \leq u_{\max} \end{aligned} \quad (2)$$

where

- $\tilde{A} \in \mathbb{R}^{n \times n}$  describes the building's thermal dynamics;
- $\tilde{x} \in \mathbb{R}^{n \times 1}$  contains the building's thermal state;
- $g(\tilde{x}, u, \Gamma)$  is a cost function that penalizes discomfort based on the occupancy  $\Gamma$  and penalizes total energy consumption;
- $u_{k \dots k+N-1}$  contains the controller output, constrained within the system's capacity  $u_{\max}$ ;
- $w_k$  is the current weather observation;
- $w_{k+1 \dots k+N-1}$  contains an up-to-date weather prediction;
- $\Gamma_k$  is the latest occupancy measurement; and
- $\Gamma_{k+1 \dots k+N-1}$  are the predicted occupancies over the horizon.

Once the optimization is solved, the first HVAC command  $u_k$  is applied, and process repeats at the next time step.

We simplify the presentation by assuming an accurate short-term weather forecast and treating the system efficiency as constant, but such assumptions are not required in practice. Where available, weather uncertainty data can be rolled into the MPC cost function in order to improve robustness [22]. Constant efficiency and the use of a single actuator simplify the actuation cost gain to a single parameter, but scenarios

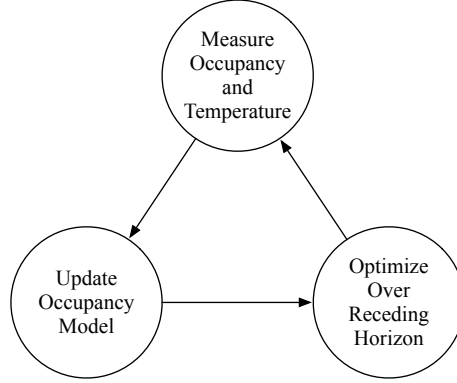


Figure 2: Process flow during operation.

with multiple actuators and correspondingly elaborate actuation cost functions have been addressed in the literature [15, 22, 12].

### 3. Building Thermal Model

Thermal model accuracy influences controller performance, so we need a thermal model that closely approximates the dominant dynamics. Here we outline how the state-space building model is generated and compare its simulation results to those obtained with EnergyPlus.

Building thermal model creation has historically been a manual process that contributes substantially to MPC implementation cost, but for this study we have used the *Sustain* platform developed at Cornell University to generate a physically representative resistor-capacitor network from a single-zone building (Figure 3) [2, 13]. The states of the thermal model are its internal temperatures, including interior wall layers and roofing materials that are not normally measured; an observer can be used to estimate those states.<sup>2</sup> Furthermore, recent literature has proposed ways to automatically tune the RC network parameters during operation and even estimate disturbances such as solar load [24].

The model under consideration has a single thermal zone and 40 other states.<sup>3</sup> It assumes well-mixed air

---

<sup>2</sup>The observability assumption is valid because of the RC network’s construction; the driving sources (exterior and interior conditions) are themselves measurable and there are no hinges in the network. See [2] for details on the RC network construction and [20, 2] Theorem 1 for a proof of observability.

<sup>3</sup>In practice, one may reduce the model size using a method like balanced truncation. We have chosen to retain the full-order model to maximize accuracy and preserve the physically intuitive state space. See [3] for a survey of reduction methods.

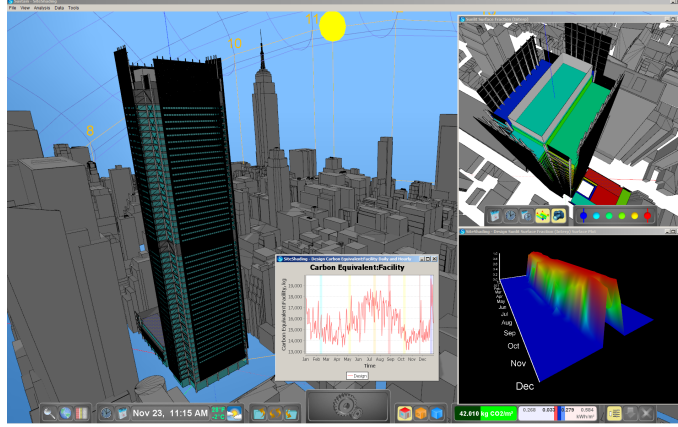


Figure 3: The *Sustain* modeling and simulation environment. (New York Times building shown.)

and uses time-invariant convection coefficients. The latter implies that the thermal gradients are always in the same direction, whereas EnergyPlus switches coefficients depending on whether the gradient enhances convection [26]. In practice, for improved accuracy, the time-invariant RC network can be adjusted before each MPC synthesis to reflect the correct coefficients or may be synthesized with time-varying coefficients. Limited support for radiant transfer has been included using coefficients adopted from EnergyPlus' *Simple* and *SimpleCombined* convection algorithms [26]. The model accepts the following inputs:

- the outside dry-bulb temperature,
- the ground temperature, and<sup>4</sup>
- heat injected by the control system.

The controller monitors a single state in the model representing the zone air temperature. The state equation of the building is

$$x_{k+1} = A_b x_k + B_w w_k + B_u u_k, \quad (3)$$

where  $k$  is the time step (in hours) and  $x_k$  is the complete temperature state vector containing the zone temperature  $x_k^{\text{zone}}$ . The vector  $w_k$  contains the weather forecast, and  $u_k$  is the heat injected into the room by the HVAC system.

<sup>4</sup>Daily ground temperature is available for free through on-line sources such as the U.S. Surface Climate Observing Reference Network [1].

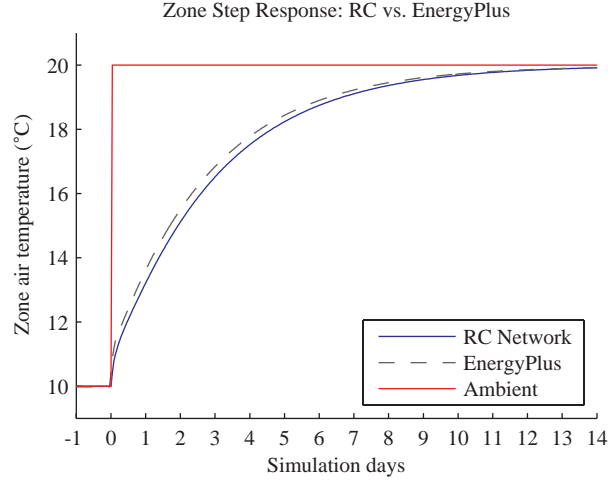


Figure 4: RC network (solid) and EnergyPlus (dashed) simulation results for a step change in ambient temperature.

Let us now validate the model by comparing the zone temperature time response of our RC network to EnergyPlus results under simplified conditions. The goal is not to exactly match EnergyPlus, but rather to show that the dominant response is plausibly close. We have simulated the same building using both the RC network and EnergyPlus under:

- a step change in air, ground, and sky infrared temperatures from 10°C to 20°C on February 1;
- no wind or humidity;
- EnergyPlus heat transfer algorithms: `Simple` convection for interior, `SimpleCombined` for exterior, and `CTF` (conduction transfer function) for wall layers.

Our RC network implementation lacks support for sky infrared transfer through windows; by matching the sky radiant temperature to the outside air temperature, we have removed this source of discrepancy from the simulation. Under the simplified conditions, very similar response times (Figure 4) suggest that the RC model is adequate for demonstration.

#### 4. Stochastic Occupancy Model

The heart of our method is its on-line trained Markov occupancy model that quickly adapts to occupancy patterns and enables the MPC to predict future occupancy. The input is a stream of asynchronous pulses from

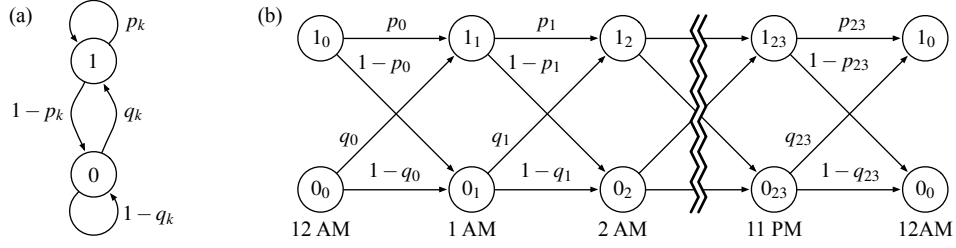


Figure 5: Occupancy model as a time-varying Markov chain (a) and unrolled into a periodic Markov chain (b).

pyroelectric infrared (PIR) or functionally similar sensors. We have chosen the Mitsubishi Electric Research Lab (MERL) motion detector data set [27], which consists of a series of one-second pulses from various motion sensors located throughout two floors of MERL and monitored over one year.<sup>5</sup> The sensors were located in hallways and conference rooms. The meetings in the Belady conference room yielded sufficiently regular occupancy patterns to showcase the benefits of MPC using on-line learning. We now discuss the Markov chain structure and then detail its training algorithm.

#### 4.1. Markov Chain Formulation

The occupancy model is as a periodic Markov chain that is updated at every observation. In its simplest form, the occupancy at time  $k$  may be in one of two states: occupied ( $\gamma_k = 1$ ) and vacant ( $\gamma_k = 0$ ). If the space is occupied at time  $k$ , there is a specific but unknown probability of remaining occupied. There is a different probability of the space becoming occupied if it is currently vacant. We wish to estimate these probabilities,

$$\begin{aligned} p_k &= \mathbb{P}[\gamma_{k+1} = 1 \mid \gamma_k = 1] \\ q_k &= \mathbb{P}[\gamma_{k+1} = 1 \mid \gamma_k = 0] \end{aligned} \quad (4)$$

using a discrete-time sequence of observations. Thus we have a two-state Markov chain with time-varying probabilities as shown in Figure 5a. The probabilities are periodic; we have chosen a period  $M = 24$  hours, so  $p_{24} \equiv p_0$  and  $q_{24} \equiv q_0$ . By unrolling the Markov chain across the day (Figure 5b) we obtain a chain with  $2M$  states, where each time instant has a  $1_k$  and  $0_k$  state, as in

$$\begin{aligned} p_k &= \mathbb{P}[\gamma_{k+1} = 1_{k+1} \mid \gamma_k = 1_k] \\ q_k &= \mathbb{P}[\gamma_{k+1} = 1_{k+1} \mid \gamma_k = 0_k] \end{aligned} \quad (5)$$

<sup>5</sup>Although we have used the MERL occupancy data, the thermal model is not one of the MERL building.



It is understood that, although the time index  $k$  in general grows without bound, the value of  $k$  is limited to the range  $0 \leq k \leq 23$  when dealing with the Markov chain, so we use  $k \bmod M$  in that case. The choice of  $M$  affects how learned patterns relate to subsequent control; if space usage patterns vary significantly across the weekdays, one might want to prevent occupancy observations on Monday from influencing control actions on Tuesday, in which case a one-week period would be more appropriate. For this study, the one-day Markov chain is trained using Monday through Friday occupancy data from the MERL data set, and weekends are not considered. In practice, one could use a separate Markov chain or switch to occupancy-reactive control during weekends.

#### 4.2. Training

A salient feature of our method is on-line occupancy model training. The training, combined with automatic forgetting, continuously refines the model without becoming rigid against shifts in the occupancy pattern. The Markov transition probabilities are backed by probability density functions that are trained iteratively using Bayes' rule.

A naive approach to predicting occupancy asks: *Given that the building was occupied (or vacant) in the previous time step, how likely is it to be occupied in the next?* Such a dichotomous perspective lacks granularity that could otherwise improve prediction accuracy. Clearly, occupancy for the majority of an hour implies different occupancy in the future compared to occupancy for just a few minutes. The question we wish to answer using the Markov chain is then: *Given a certain percentage of occupancy over the previous hour, how much occupancy do we expect in the future?* We approach the problem in three steps. First, we explain the simplest case where boolean occupancy observations are used to train the Markov chain directly. Next, we augment the boolean training with a forgetting factor to accommodate changes to occupancy patterns. Finally, we refine the approach to use fractional occupancy in order to improve prediction accuracy.

##### *Observed boolean occupancy*

Each state of the Markov chain has two possible transition paths; the problem of estimating transition probabilities is loosely analogous to estimating the biases of two coins. The well-known probability function of a biased coin tossed  $N$  times is

$$f(\theta, N, N_H) = \binom{N}{N_H} \theta^{N_H} (1 - \theta)^{N - N_H} \quad (6)$$

where  $\binom{N}{N_H}$  is the number of ways to permute  $N_H$  heads in a sequence of  $N$  tosses and  $\theta$  is the probability of heads for each toss. With  $\theta$  and  $N$  fixed,  $f$  is a binomial distribution on  $N_H$ ; with  $N$  and  $N_H$  fixed,  $f$  is a continuous beta distribution<sup>6</sup> over  $\theta$ , where  $\int_0^1 f(\theta) d\theta = 1$ . Another way to arrive at this distribution is to iteratively apply Bayes' rule each time the coin is tossed using

$$f_j(\theta \mid x_{1\dots j}) \sim \theta^{x_j=H} (1-\theta)^{x_j \neq H} f_{j-1}(\theta \mid x_{1\dots j-1}) \quad f_0(\theta) = 1 \quad (7)$$

where  $x_j$  is the latest toss outcome. The function  $\theta^{x_j=H} (1-\theta)^{x_j \neq H}$  is the likelihood of a particular probability  $\theta$  given an observed outcome  $x_j$ . (The expression  $x_j = H$  is 1 if true and 0 if false.) The likelihood function is multiplied by the prior distribution  $f_{j-1}(\theta)$ , which is initially a uniform distribution; when no information is known about  $\theta$ , we assume all values are equally likely—although in practice  $f_0$  may be any distribution that represents the initial knowledge. The  $\sim$  notation means that, after multiplication, the function is scaled so that the necessary condition  $\int_0^1 f_j(\theta \mid x_j) d\theta = 1$  holds.

Coin tosses are independent, but occupancy transitions are not; thus, instead of one distribution, we have to maintain two for each time step—one over transitions from the occupied state, and the other for transitions out of the vacant state. Let  $\gamma_k \in \{0, 1\}$  be the occupancy state at time  $k$ . Then the transition probabilities of interest are

$$\begin{aligned} p_k &= \mathbb{P}[\gamma_{k+1} = 1 \mid \gamma_k = 1] \\ q_k &= \mathbb{P}[\gamma_{k+1} = 1 \mid \gamma_k = 0], \end{aligned} \quad (8)$$

with probability density functions  $f(p_k)$  and  $g(q_k)$ . The index  $j$  increments each time a particular distribution is trained. The functions  $f(p_k)$  and  $g(q_k)$  are updated using

$$\begin{aligned} f_j(p_k \mid \gamma_{k+1} = 1, \gamma_k = 1) &\sim p_k^{\gamma_{k+1}=1} (1-p_k)^{\gamma_{k+1} \neq 1} f_{j-1}(p_k) \\ f_j(p_k \mid \gamma_{k+1} = 1, \gamma_k = 0) &= f_{j-1}(p_k) \\ g_j(q_k \mid \gamma_{k+1} = 1, \gamma_k = 0) &\sim q_k^{\gamma_{k+1}=1} (1-q_k)^{\gamma_{k+1} \neq 1} g_{j-1}(q_k) \\ g_j(q_k \mid \gamma_{k+1} = 1, \gamma_k = 1) &= g_{j-1}(q_k) \\ f_0(p_k) &= g_0(q_k) = 1 \end{aligned} \quad (9)$$

---

<sup>6</sup>Although individual distributions on the transition probability are of the beta type, they become complex piecewise distributions (that are algorithmically cumbersome to maintain) when used in linear combination with each other. In general, sums of beta distributions are not themselves beta distributions [14].

where the  $\sim$  notation means that the results have been scaled so that  $\int_0^1 f_j(p_k) dp_k = 1$  and  $\int_0^1 g_j(q_k) dq_k = 1$ . The distribution  $f_j(p_k)$  does not change from  $f_{j-1}(p_k)$  if the space was vacant at step  $k$ ; similarly, if the space was occupied, we leave  $g_j(q_k) = g_{j-1}(q_k)$  because no transition out of  $0_k$  was observed.

### *Forgetting Factor*

As training proceeds, the distributions  $f(p_k)$  and  $g(q_k)$  become increasingly narrow and converge toward delta functions. The oldest and newest training data exert equal influence, which means that subsequent training becomes increasingly diluted across an unlimited history. This is fine for batch training, but problematic under continuous on-line training, because the distributions become immutable and sometimes numerically unstable after many iterations. The forgetting factor  $\lambda$  gives progressively smaller weight to old training data and allows the Markov chain to retain its flexibility. Linear forgetting is implemented using

$$\begin{aligned} f'_j(p_k) &= \lambda f_j(p_k) + (1 - \lambda) f_0(p_k) \\ g'_j(q_k) &= \lambda g_j(q_k) + (1 - \lambda) g_0(q_k) \end{aligned} \tag{10}$$

where  $f_0$  and  $g_0$  are the initial uniform distributions, and  $f_j$  and  $g_j$  are the posterior distributions that have just been trained but without forgetting applied.<sup>7</sup>

To illustrate the effect of forgetting on the distributions, we have trained a single state of the Markov chain repeatedly using alternating transitions  $\gamma_0 = 0 \rightarrow \gamma_1 = 1$  and  $\gamma_0 = 0 \rightarrow \gamma_1 = 0$ . This is analogous to flipping an unbiased coin numerous times and observing half heads and half tails, so we expect an increasingly narrow distribution on  $p_0$  with a peak near 0.5. Figure 6a shows how the distribution evolves from the initial flat line  $f_0(p_0) = 1$  toward its final shape  $f_{100}(p_0)$ . This result would be preferred if the transition pattern was never expected to change, but such a concentrated distribution prevents subsequent adaptation.

The result with forgetting is a distinctly broader distribution. Figure 6b shows the result of the same training sequence but with 15% forgetting ( $\lambda = 0.85$ ). The distribution—and therefore its expected value—shifts laterally with each alternate observation, even after many training iterations, and remains off the horizontal axis; this extra mobility reflects greater adaptability.

Forgetting applied this way also affects the training retention time. With conventional, off-line batch training, all the training data comes from a hand-picked set, typically from a recent fixed time interval. The training data set may not, however, contain all possible transitions; some transitions may be observed an insignificant number of times or not at all. On-line training with forgetting alleviates this problem, because

---

<sup>7</sup>This is but one of several ways to implement forgetting; see [17] for a survey.

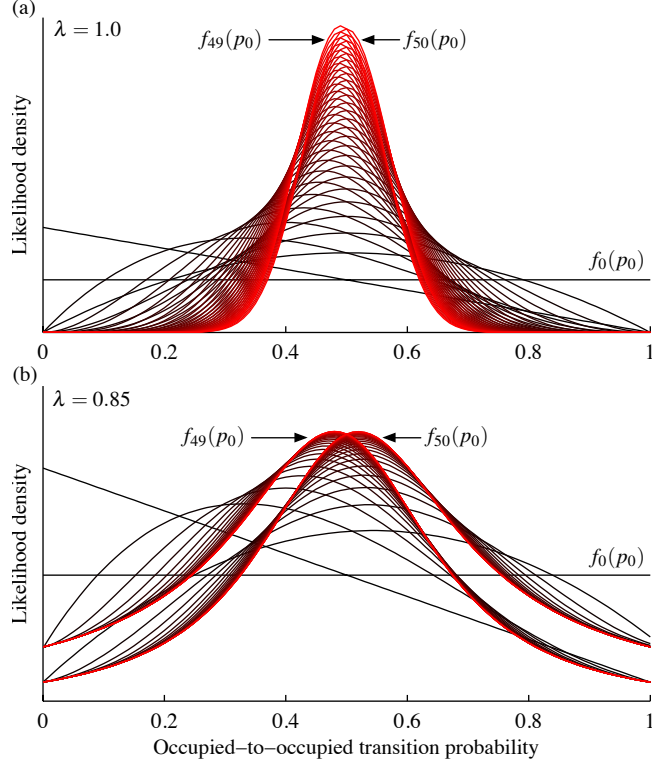


Figure 6: A particular occupancy transition probability distribution when trained with alternating outcomes data using no forgetting ( $\lambda = 1$ ) (a) and with partial forgetting ( $\lambda = 0.85$ ) (b). The forgetting factor broadens the distribution and allows it to shift laterally even when extensively trained, reflecting greater ability to adjust to changes in space usage. For each case, the initial distribution is uniform (black horizontal trace). Higher levels of training are shown as brighter color.

information is forgotten only when a new observation is made. Therefore, infrequently observed patterns are retained longer than they would be under batch training. Of course, forgetting may also be applied across all transitions if this effect is not desired.

#### *Using Fractional Occupancy*

The occupancy observer and Markov model are fed by PIR sensors that asynchronously indicate movement in a space but that cannot positively identify vacancy. The controller, in contrast, requires a discrete-time series of occupancy percentages. A simple heuristic converts the asynchronous pulses (Figure 7a) to the necessary discrete signal in two steps. First, a dwell time bridges most consecutive pulses without significantly inflating the computed occupancy periods, yielding a continuous-time boolean signal  $\gamma(t)$  (Figure 7b).

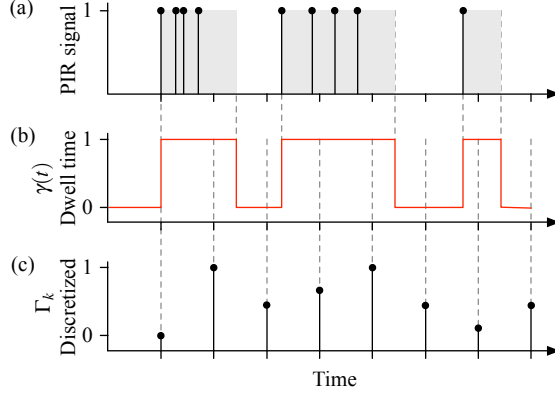


Figure 7: Asynchronous sensor pulses (a); Derived continuous signal using dwell time (b); Resulting discrete-time occupancy percentage (c).

A discretization step then superimposes a fixed time grid onto the asynchronous signal. Instead of sampling  $\gamma_k \in \{0, 1\}$  at each time step, we average occupancy over the interval to obtain

$$\Gamma_k = \frac{1}{t_k - t_{k-1}} \int_{t_{k-1}}^{t_k} \gamma(t) dt \equiv \mathbb{P}[\gamma_k = 1] \in [0, 1] \quad (11)$$

as in Figure 7c. Thus the occupancy measurement admits a continuous range of values. We subsequently refer to the signal  $\gamma_k$  as an unobservable boolean occupancy that is instead viewed probabilistically through  $\Gamma_k$  according to a distribution  $\pi_k$ . An observation  $\Gamma_k = 60\%$  is treated equivalently to  $\mathbb{P}[\gamma_k = 1] = 0.6$ . From this we estimate the occupancy at the next time step using

$$\begin{aligned} \mathbb{P}[\Gamma_{k+1} = 1 \mid \Gamma_k] &= \Gamma_k \mathbb{P}[\gamma_{k+1} = 1 \mid \gamma_k = 1] \\ &\quad + (1 - \Gamma_k) \mathbb{P}[\gamma_{k+1} = 1 \mid \gamma_k = 0] \\ &= \Gamma_k \mathbb{E}[p_k] + (1 - \Gamma_k) \mathbb{E}[q_k] \end{aligned} \quad (12)$$

where the expectation operator reflects the fact that  $p_k$  and  $q_k$  are not directly known but are instead estimated through Bayesian inference. At each step  $k$ , there are four possible state transitions with associated posterior distributions

$$\begin{aligned} \gamma_k = 1 \rightarrow \gamma_{k+1} = 1 : & \quad f_j^{(1)}(p_k) \sim p_k f_{j-1}(p_k) \\ \gamma_k = 1 \rightarrow \gamma_{k+1} = 0 : & \quad f_j^{(0)}(p_k) \sim (1 - p_k) f_{j-1}(p_k) \\ \gamma_k = 0 \rightarrow \gamma_{k+1} = 1 : & \quad g_j^{(1)}(q_k) \sim q_k g_{j-1}(q_k) \\ \gamma_k = 0 \rightarrow \gamma_{k+1} = 0 : & \quad g_j^{(0)}(q_k) \sim (1 - q_k) g_{j-1}(q_k) \end{aligned} \quad (13)$$

where

- $f_j^{(1)}$  is the updated posterior distribution as if  $\gamma_k = 1$  and  $\gamma_{k+1} = 1$  had been observed,
- $f_j^{(0)}$  is similar to  $f_j^{(1)}$  but updated as if  $\gamma_{k+1} = 0$  had been observed,
- $g_j^{(1)}$  is the updated posterior distribution as if  $\gamma_k = 0$  and  $\gamma_{k+1} = 1$  had been observed, and
- $g_j^{(0)}$  is similar to  $g_j^{(1)}$  but updated as if  $\gamma_{k+1} = 0$  had been observed.

To obtain  $f_j(p_k)$ , we blend  $f_j^{(1)}(p_k)$  and  $f_j^{(0)}(p_k)$  according to the fractional observation  $\Gamma_{k+1}$ . We then weight the training according to the existing observation  $\Gamma_k$ . The training for  $g_j(q_k)$  follows analogously.

$$\begin{aligned}
f_j(p_k) &= \Gamma_k \left( \Gamma_{k+1} f_j^{(1)}(p_k) + (1 - \Gamma_{k+1}) f_j^{(0)}(p_k) \right) \\
&\quad + (1 - \Gamma_k) f_{j-1}(p_k) \\
g_j(q_k) &= (1 - \Gamma_k) \left( \Gamma_{k+1} g_j^{(1)}(q_k) + (1 - \Gamma_{k+1}) g_j^{(0)}(q_k) \right) \\
&\quad + \Gamma_k g_{j-1}(q_k)
\end{aligned} \tag{14}$$

Values of  $\Gamma_k$  closer to one bias the training more heavily toward  $f(p_k)$ , and values closer to zero cause more training of  $g(q_k)$ .

Training data patterns influence the shapes of  $f(p_k)$  and  $g(q_k)$ . In Figure 8a we see that vacancy at 2:00am strongly implies vacancy at 3:00am, and occupancy at 2:00am weakly implies occupancy at 3:00am. In other words, early morning visitors rarely stay for very long, and the  $\gamma_2 = 0_2 \rightarrow \gamma_3 = 1_3$  distribution (dashed curve) is accordingly heavily biased toward zero. Because occupancy is rare at 2:00am, the transition  $\gamma_2 = 1_2 \rightarrow \gamma_3 = 1_3$  is very weakly observed and yields a much flatter distribution. Occupancy at 3:00pm (Figure 8b) is more varied, resulting in more balanced training and similarly shaped distributions. The distributions reveal that occupancy at 3:00pm implies occupancy at 4:00pm, but meetings are unlikely to start after 3:00pm.

Forgetting is applied similarly to Equation 10, where Equations 14 are used instead for the posterior distributions.

#### 4.3. Transition Matrix and Occupancy Prediction

In order to predict future occupancy, we need the Markov transition probability matrix  $P$  shown in Equation 1. Recall from Section 4.1 and Figure 5b that the two-state Markov chain has been unrolled into a 48-state chain with states  $1_0 \dots 1_{23}$  and  $0_0 \dots 0_{23}$ . The matrix  $P$  can be constructed from the four sub-matrices

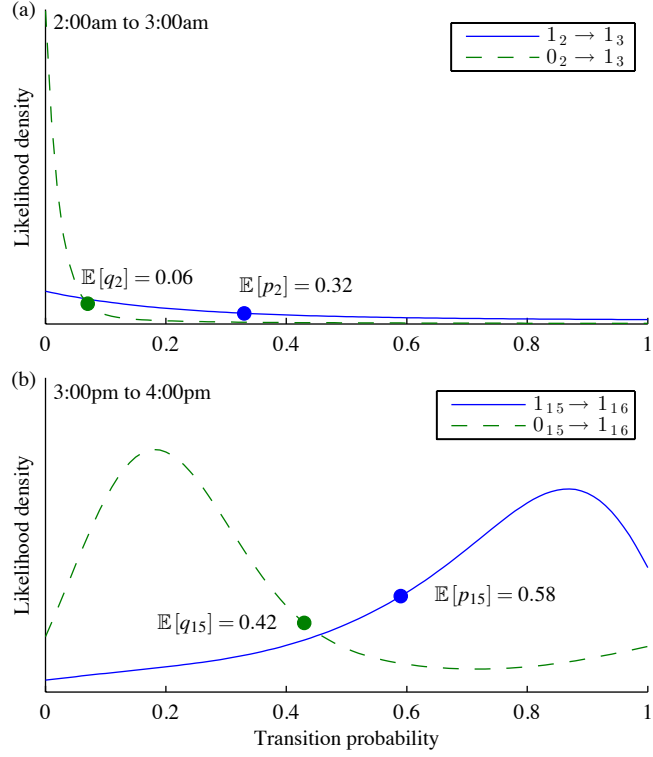


Figure 8: Probability densities for the occupied-to-occupied (solid) and vacant-to-occupied (dashed) at 2:00am (a) and 3:00pm (b). Occupancy is rare at 2:00am, so the  $0_2 \rightarrow 1_3$  distribution is heavily biased toward zero probability of future occupancy. The training data for 3:00pm is marked by greater variety.

- $P^{(\text{I})}$  for  $1_k \rightarrow 1_{k+1}$  transitions,
- $P^{(\text{II})}$  for  $1_k \rightarrow 0_{k+1}$  transitions,
- $P^{(\text{III})}$  for  $0_k \rightarrow 1_{k+1}$  transitions, and
- $P^{(\text{IV})}$  for  $0_k \rightarrow 0_{k+1}$  transitions.

The entries for  $P^{(\text{I})}$  and  $P^{(\text{IV})}$  are the expected values over distributions  $f(p)$  and  $g(q)$ , and the other two matrices are their complements.

$$\begin{aligned}
P_{ik}^{(\text{I})} &= \begin{cases} \mathbb{P}[\gamma_k = 1 \mid \gamma_i = 1] = \mathbb{E}[p_i] & (*) \\ 0 & \text{otherwise} \end{cases} \\
P_{ik}^{(\text{II})} &= \begin{cases} \mathbb{P}[\gamma_k = 0 \mid \gamma_i = 1] = 1 - \mathbb{E}[p_i] & (*) \\ 0 & \text{otherwise} \end{cases} \\
P_{ik}^{(\text{III})} &= \begin{cases} \mathbb{P}[\gamma_k = 1 \mid \gamma_i = 0] = \mathbb{E}[q_i] & (*) \\ 0 & \text{otherwise} \end{cases} \\
P_{ik}^{(\text{IV})} &= \begin{cases} \mathbb{P}[\gamma_k = 0 \mid \gamma_i = 0] = 1 - \mathbb{E}[q_i] & (*) \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

(\*)  $k = i + 1 \pmod{M}$  (15)

The matrix  $P^{(\text{I})}$ , for example, takes the form

$$P^{(\text{I})} = \mathbb{E} \begin{bmatrix} 0 & p_{01} & & & \\ & 0 & p_{12} & & \\ & & \ddots & \ddots & \\ & & & 0 & p_{N-2} \\ p_{N-1} & & & & 0 \end{bmatrix} \quad (16)$$

where we have skirted convention by calling the top-left element  $P_{00}$  instead of  $P_{11}$ . The complete Markov transition matrix is

$$P = \left[ \begin{array}{c|c} P^{(\text{I})} & P^{(\text{II})} \\ \hline P^{(\text{III})} & P^{(\text{IV})} \end{array} \right]. \quad (17)$$



Recalling Equation 1, the expected occupancy  $m$  steps in the future given a current estimate  $\Gamma_k$  is

$$\mathbb{E}[\Gamma_{k+m} | \Gamma_k] = \underbrace{\begin{bmatrix} \Gamma_k \mathbb{1}_{1 \times M} & (1 - \Gamma_k) \mathbb{1}_{1 \times M} \end{bmatrix}}_{\pi_k} P^m \begin{bmatrix} 1_{M \times 1} \\ 0_{M \times 1} \end{bmatrix} \quad (18)$$

where  $\mathbb{1}_{1 \times M}^k$  is a vector with the  $k$ th element set to one and all others left zero.

## 5. MPC Formulation

To balance competing demands for occupant comfort and low total energy consumption, we need to avoid conditioning the space when it is expected to be vacant; the level of comfort should be maintained based on immediate and predicted occupancy. To simplify the definition of the cost function, we have augmented the building's state space model to include the non-changing comfort temperature setpoint and a weather forecast shift-register system, i.e.

$$\underbrace{\begin{bmatrix} x_{k+1} \\ \tau_{k+1} \\ \phi_{k+1} \end{bmatrix}}_{\tilde{x}_{k+1}} = \tilde{A} \underbrace{\begin{bmatrix} x_k \\ \tau_k \\ \phi_k \end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix} B_u \\ 0 \\ 0 \end{bmatrix}}_{\tilde{B}} u_k \quad (19)$$

where  $x$  is the building's thermal state (Equation 3),  $\tau$  is the comfort setpoint, and  $\phi$  is a shift-register state that iterates through the weather forecast over the MPC horizon. The augmented matrix  $\tilde{A}$  connects the weather forecast to the building thermal model internally.<sup>8</sup> At each instant  $k$ , we seek the optimal control law

$$\begin{aligned} u_k^*(\tilde{x}_k, \Gamma_k) = & \arg \min_u \mathbb{E} \left[ \sum_{j=0}^{N-1} g(\tilde{x}_{j+k}, u_{j+k}, \Gamma_{j+k}) \right] \\ & \text{subject to} \quad \tilde{x}_{i+1} = \tilde{A} \tilde{x}_i + \tilde{B} u_i \quad \forall i \in \mathbb{Z}^+ \\ & \quad \quad \quad 0 \leq u \leq u_{\max} \end{aligned} \quad (20)$$

where  $u_{j+k}$  is an individual control action and  $\Gamma_{j+k} \in [0, 1]$  is either an occupancy measurement or an occupancy prediction. This is standard except that the stage cost  $g$  adjusts the discomfort weighting factor based

---

<sup>8</sup>Because the forecast is updated at each time step, our implementation adjusts the augmented transition matrix  $\tilde{A}$  before each MPC synthesis to reflect the latest prediction. This simplifies the cost function and allows the MPC to be formulated in a compact vectorized form as detailed in Equation 3.8 of [25].

on occupancy, in our case

$$\begin{aligned} g(\tilde{x}, u, \Gamma) &= \tilde{x}^\top \Gamma Q \tilde{x} + r |u| \\ &= \Gamma \beta (x_{\text{zone}} - \tau)^2 + r |u| \end{aligned} \quad (21)$$

where

- $\tilde{x}$  is the augmented system state vector and  $x_{\text{zone}}$  is the particular zone air temperature under control,
- $\tau$  is the comfortable setpoint temperature,
- $Q$  is a matrix that extracts  $\beta(x_{\text{zone}} - \tau)^2$  from  $\tilde{x}^\top Q \tilde{x}$ ,
- $\beta$  is the maximum discomfort cost gain,
- $\Gamma$  is the observed or predicted occupancy,
- $r$  is a scalar constant to penalize energy use<sup>9</sup>, and
- $u$  is the heat input to the zone.

We have chosen to penalize  $|u|$ , rather than  $u^2$ , because quadratic cost suppresses peaks and spreads control action over time; peak suppression inhibits the full system shutdown necessary to save energy during vacancy.

The many ( $2^N$ ) possible occupancy state trajectories, along with the constraints on  $u$ , make it difficult to find a closed-form solution using exact dynamic programming. (Recall from Figure 5 that each occupancy state has two possible outgoing transitions.) If we instead condition all occupancy predictions solely on the present observation, we obtain the approximate program

$$\begin{aligned} u_k^*(\tilde{x}_k, \Gamma_k) &\approx \arg \min_{0 \leq u \leq u_{\max}} \left\{ g(\tilde{x}_k, u_k, \Gamma_k) \right. \\ &\quad \left. + \sum_{j=1}^{N-1} g(\tilde{x}_{j+k}, u_{j+k}, \mathbb{E}[\Gamma_{j+k} | \Gamma_k]) \right\}, \end{aligned} \quad (22)$$

---

<sup>9</sup>In this simplified formulation, only the ratio of  $r$  and  $\beta$  matters; together, they constitute a single tuning adjustment.

where the calculation for  $\mathbb{E}[\Gamma_{j+k} | \Gamma_k]$  comes from Equation 18.<sup>10</sup> The optimization is then

$$\begin{aligned} \min_{u_k \dots u_{k+N-1}} \quad & \sum_{j=0}^{N-1} \tilde{x}_{j+k}^\top \mathbb{E}[\Gamma_{j+k} | \Gamma_k] Q \tilde{x}_{j+k} + |u_{j+k}| \\ \text{subject to} \quad & \tilde{x}_{i+1} = \tilde{A} \tilde{x}_i + \tilde{B} u_i \quad \forall i \in \mathbb{Z}^+ \\ & 0 \leq u \leq u_{\max} \end{aligned} \tag{23}$$

As with conventional MPC, the controller applies  $u_k$  to the system and discards  $u_{k+1} \dots u_{k+N-1}$ ; the solution is repeated at each subsequent step.

## 6. Comparison to Conventional Control

To demonstrate the algorithm's advantages over conventional control, we have run a simulation under the following conditions:

- MERL occupancy data for the Belady conference room (sensors 452 and 453) from March through June 2006 for training;
- MERL occupancy data for June 7–9, 2006 for simulation;
- EnergyPlus weather data for Elmira, NY for May 10–12 (typical meteorological year);
- no un-modeled disturbances;
- one-hour time step;
- forgetting factor  $\lambda = 95\%$ ;
- warm-up period of 21 days with  $x_{\text{zone}} = 23^\circ\text{C}$ ;
- system capacity of 7.5kW.

Figure 9 shows simulation results for three identically-tuned MPC implementations:

1. a purely occupancy-reactive controller that starts the system only after detecting occupancy,
2. a conservatively scheduled setback controller supplemented with occupancy reaction, and

---

<sup>10</sup>Multi-parametric methods can be used to partition the state space into regions, each with an exact control law parameterized on the entire state at the expense of a more complex MPC formulation [7].

3. an occupancy-predictive controller trained with the preceding weeks' data.

A dominant 7:00am–8:00pm occupancy pattern is evident in Figure 9b (solid black line) with significant fluctuation during the workday. The reactive-only controller (■) is designed to regulate the temperature to 23°C during occupied hours and switch off during vacant hours. It is no surprise that this controller yields the lowest energy consumption but poor comfort performance in the early morning (7:00am–9:00am) due to thermal lag and system capacity limitation. (Thermal lag becomes an even more important consideration for systems with significantly longer warm-up times, such as in-slab radiant loops.) The second baseline controller (▼) activates during occupancy and follows a conservatively set (5:00am–8:00pm) schedule to mitigate the effects of thermal lag in the morning; this controller ensures maximum occupant comfort but also consumes the most energy.

The predictive controller (●) performs between these bounds. Having been trained from previous weeks' occupancy patterns, it is able to correctly estimate occupants' arrival time and preheats the space slightly later than the scheduled controller. It also allows the temperature to droop toward the end of the day, thus increasing average discomfort over the day but saving substantial energy—approximately 23kWh (Table 1) compared to the scheduled controller. An unexpected two-hour occupancy at 11:00pm is weakly anticipated at by the controller, and the system responds less aggressively than the reactive and scheduled controllers. The predictive controller repeats the pattern on subsequent days, but progressively subdued, thus demonstrating that it has learned the pattern and then gradually forgets it when it fails to reappear. Even with the incorrect predictions on the second and third nights, the total energy consumption is lower than the scheduled controller. The automatic training and forgetting means that the system will continue to produce good performance without ongoing manual adjustment. (A fixed schedule may, of course, supplement or override the occupancy predictions as needed.)

## 7. Conclusion

We have demonstrated the use of model predictive control with a stochastic occupancy model to reduce HVAC energy consumption using both occupancy reaction and prediction. The occupancy has been modeled as a Markov chain trained automatically during operation. The implementation is simplified by approximate dynamic programming whereby building occupancy is projected multiple steps into the future using current occupancy observations. We remark that although our method relies on weather forecasts and a dynamical model of the building, on-line data sources and emerging software tools can remove these implementation

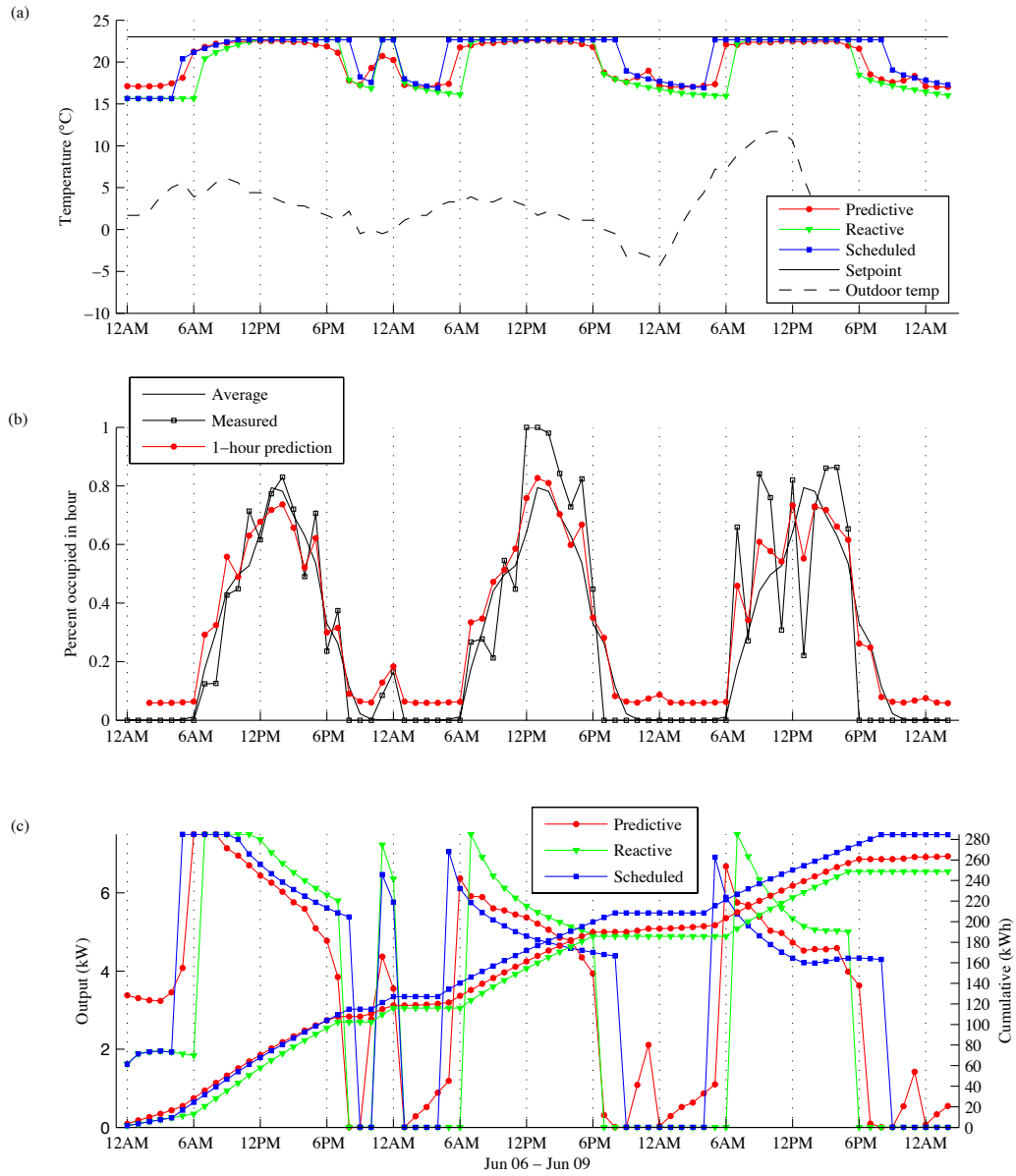


Figure 9: Simulation results for a single-zone building. The Markov model partially predicts the occupancy at 11:00pm and begins to ignore it on subsequent days, thus illustrating the effect of the forgetting factor  $\lambda$ . Temperature control performance is shown in (a), one hour occupancy prediction is shown in (b), and instantaneous and cumulative energy consumption are shown in (c).

	Discomfort $ T_{\text{zone}} - \tau $ ( $^{\circ}\text{C}$ rms)			Energy Usage (kWh)		
	Pred.	Reac.	Sched.	Pred.	Reac.	Sched.
Day 1	0.428	0.773	0.427	114	110	121
Day 2	0.396	0.756	0.199	78.2	76.1	87.1
Day 3	0.337	1.18	0.190	69.5	62.9	76.0
Overall	0.381	0.904	0.259	262	249	285
Peak	4.39	5.82	2.09			

Table 1: Performance of predictive, reactive, and scheduled algorithms over a three-day period. The reactive controller yields the lowest energy consumption but worst rms and peak discomfort, while the scheduled controller uses the most energy to provide the most comfortable conditions. The predictive algorithm falls within this envelope with mid-range comfort performance and energy consumption similar to that of reactive control.

obstacles in the near future.

We have made some simplifications to improve clarity. First, we have chosen a rather coarse one-hour time step, even though practical controllers normally operate on a much finer time scale to provide adequate bandwidth; however, the Markov model may operate on an entirely different time scale from the MPC with only minor implementation changes. Second, our hypothetical system has constant efficiency and operates only in heat mode; these were chosen in order to simplify the cost function to minimize distraction from the contribution. So long as energy consumption can be estimated and room temperature can be measured, the stochastic occupancy model may be added to a more complex MPC. Finally, to improve performance, we have incorporated weather observations and predictions into our model under an assumption of accuracy; recent research has addressed ways to incorporate uncertainty into the optimization where additional robustness is desired. Demonstrating our algorithm on a full-scale system without these simplifications is left to future work.

## 8. Acknowledgements

The authors thank Peter Radecki for providing constructive feedback throughout this work.

## 9. References

### References

- [1] National Climactic Data Center. U.S. surface climate observing reference network. <https://www.ncdc.noaa.gov/crn/qcdatasets.html>, February 2014.
- [2] Justin R. Dobbs and Brandon M. Hincey. Automatic model reduction in architecture: a window into building thermal structure. In *Proceedings of the SimBuild, 5th National Conference of IBPSA-USA, Madison, WI*. IBPSA, 2012.
- [3] Justin R. Dobbs and Brandon M. Hincey. A comparison of thermal zone aggregation methods. In *51st IEEE Conference on Decision and Control*, volume 1, pages 6938–6944. IEEE, 2012.
- [4] Robert H Dodier, Gregor P Henze, Dale K Tiller, and Xin Guo. Building occupancy detection through sensor belief networks. *Energy and buildings*, 38(9):1033–1043, 2006.
- [5] Bing Dong and Khee Poh Lam. A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. In *Building Simulation*, volume 7, pages 89–106. Springer, 2014.
- [6] Bing Dong, Khee Poh Lam, and C Neuman. Integrated building control based on occupant behavior pattern detection and local weather forecasting. In *Twelfth International IBPSA Conference. Sydney: IBPSA Australia*, pages 14–17, 2011.
- [7] Vivek Dua Efstratios N. Pistikopoulos, Michael C. Georgiadis. *Multi-Parametric Programming: Theory, Algorithms, and Applications*, volume 1. Wiley-VCH Verlag, 2007.
- [8] Varick L Erickson, Miguel Á Carreira-Perpiñán, and Alberto E Cerpa. Observe: Occupancy-based system for efficient reduction of hvac energy. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 258–269. IEEE, 2011.
- [9] Varick L Erickson and Alberto E Cerpa. Occupancy based demand response HVAC control strategy. In *Proceedings of the Second ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 7–12. ACM, 2010.

- [10] Varick L Erickson, Yiqing Lin, Ankur Kamthe, Rohini Brahme, Amit Surana, Alberto E Cerpa, Michael D Sohn, and Satish Narayanan. Energy efficient building environment control strategies using real-time occupancy measurements. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 19–24. ACM, 2009.
- [11] Ge Gao and Kamin Whitehouse. The self-programming thermostat: Optimizing setback schedules based on home occupancy patterns. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pages 67–72, New York, NY, USA, 2009. ACM.
- [12] Siddharth Goyal, Herbert A Ingle, and Prabir Barooah. Occupancy-based zone-climate control for energy-efficient buildings: Complexity vs. performance. *Applied Energy*, 106:209–221, 2013.
- [13] Donald Greenberg, Kevin Pratt, Brandon Hincey, Nathaniel Jones, Lars Schumann, Justin Dobbs, Zhao Dong, David Bosworth, and Bruce Walter. Sustain: An experimental test bed for building energy simulation. *Energy and Buildings*, 58(0):44 – 57, 2013.
- [14] Arjun Kumar Gupta and Saraleesan Nadarajah. *Handbook of beta distribution and its applications*, volume 175, pages 69–84. CRC Press, 2004.
- [15] Phillip Haves, Brandon Hincey, Francesco Borrell, John Elliot, Yudong Ma, Brian Coffey, Sorin Benghea, and Michael Wetter. Model predictive control of HVAC systems: Implementation and testing at the University of California, Merced. Technical report, Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US), 2010.
- [16] J. Hutchins, A. Ihler, and P. Smyth. Modeling count data from multiple sensors: A building occupancy model. In *Computational Advances in Multi-Sensor Adaptive Processing, 2007. CAMPSAP 2007. 2nd IEEE International Workshop on*, pages 241–244, Dec 2007.
- [17] R. Kulhavý and M. B. Zarrop. On a general concept of forgetting. *International Journal of Control*, 58(4):905–924, 1993.
- [18] Khee Poh Lam, Michael Höynck, Bing Dong, Burton Andrews, Y Chiou, Rui Zhang, Diego Benitez, Joonho Choi, et al. Occupancy detection through an extensive environmental sensor network in an open-plan office building. *IBPSA Building Simulation*, pages 1452–1459, 2009.



- [19] Chenda Liao, Yashen Lin, and Prabir Barooah. Agent-based and graphical modelling of building occupancy. *Journal of Building Performance Simulation*, 5(1):5–25, 2012.
- [20] Kai-Sheng Lu and Kunsheng Lu. Controllability and observability criteria of RLC networks over  $F(z)$ . *International Journal of Circuit Theory and Applications*, 29(3):337–341, 2001.
- [21] Sean Meyn, Amit Surana, Yiqing Lin, Stella M Oggianu, Satish Narayanan, and Thomas A Frewen. A sensor-utility-network method for estimation of occupancy in buildings. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 1494–1500. IEEE, 2009.
- [22] Frauke Oldewurtel, Alessandra Parisio, Colin N. Jones, Dimitrios Gyalistras, Markus Gwerder, Vanessa Stauch, Beat Lehmann, and Manfred Morari. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45(0):15 – 27, 2012.
- [23] Jessen Page, Darren Robinson, Nicolas Morel, and J-L Scartezzini. A generalised stochastic model for the simulation of occupant presence. *Energy and Buildings*, 40(2):83–98, 2008.
- [24] Peter Radecki and Brandon M. Hencely. Online thermal estimation, control, and self-excitation of buildings. In *Proceedings of IEEE Conference on Decision and Control*, Firenze, Italy, December 2013. IEEE.
- [25] J Anthony Rossiter. *Model-based predictive control: a practical approach*, chapter 3.2. CRC press, Boca Raton, 2003.
- [26] United States Department of Energy. *EnergyPlus Engineering Document: The Reference to Energy-Plus Calculations*. University of Illinois and University of California, 2011.
- [27] Christopher R. Wren, Yuri A. Ivanov, Darren Leigh, and Jonathan Westhues. The MERL motion detector dataset. In *Proceedings of the 2007 Workshop on Massive Datasets*, MD '07, pages 10–14, New York, NY, USA, 2007. ACM.